INFORMATION AND CONTROL 62, 64–92 (1984)

Optimal Hashing

R. E. KRICHEVSKY

Mathematical Institute, Academy of Sciences, Novosibirsk 630090, USSR

A concept of complexity of hashing is introduced and studied with special attention to the lower bounds of complexity. A new class of rather simple hash-functions is developed. These functions are shown to be near optimal within this concept of complexity. © 1984 Academic Press, Inc.

1. INTRODUCTION

1.1. *Statement of the Problem*

The question under consideration is: how to choose a good hash-function? First, some definitions. Let n and m be natural, E^n be the set of all n-length binary words, D be a dictionary, i.e., a subset of E^n, $|D|$ be the cardinality of D. A map f from E^n to $\{0, 1,..., m-1\}$ is called word-address (or key-address) map, $f(x)$ is called the address of x, $x \in D$, $\alpha = |D|/m$ is called loading factor. The number m is the range of f on E^n. The sets $\{f^{-1}(k) \cap D\}$ are called clusters, the vector i whose coordinates are the cardinalities of clusters is called the signature of f on D,

$$i = i_f(D) = (i_0,..., i_{m-1}), \qquad i_k = |f^{-1}(k) \cap D|, k = 0,..., m-1.$$

Obviously,

$$|i| = \sum_{k=0}^{m-1} i_k = |D|. \tag{1.1}$$

The number

$$I(f, D) = \frac{1}{|D|} \sum_{k=0}^{m-1} i_k^2 - 1 \tag{1.2}$$

is called noninjectivity index, or, briefly, index of a map f on a dictionary D. If f is an injection, then $I(f, D)$ attains its minimum 0; if f is "the most noninjective map," i.e., its range equals 1, then $I(f, D)$ attains its maximum $|D| - 1$. So, the index indicates how far from an injection a map f is. But it has one more meaning. Suppose the words of a dictionary D to be loaded

0019-9958/84 $3.00

into a computer. The kth cluster of D forms an i_k-length list, $k = 0,..., m-1$. The lists are separated from each other. It takes one unit of time to find the first word of the kth list,..., i_k units of time to find the last one. The total search time equals $\sum_{k=0}^{m-1} 1 + \cdots + i_k = \sum_{k=0}^{m-1} i_k(i_k+1)/2$, the average search time $t(f, D) = (1/|D|)\sum_{k=0}^{m-1} i_k(i_k+1)/2$, if all words have the same probability. One can find the last formula in Knuth (1973). The numbers $t(f, D)$ and $I(f, D)$ are linearly dependent:

$$t(f, D) = \tfrac{1}{2}I(f, D) + 1. \tag{1.3}$$

It is more convenient to deal with the index $I(f, D)$ rather than with the average time $t(f, D)$. But it is easy to restate the results for $t(f, D)$ via (1.3).

There are two kinds of methods to load a dictionary into computer memory and then to search for a word. For the first of them the index of the word-address map f equals zero. They are injective or collision-free. For the second ones the index is positive. They are called hash-methods, the map f is called a hash-function, $f(x)$ is called hash-adress of a word x.

Keeping words with the same hash-adresses on separate lists as described above is called separate chaining. Apart from that, there are other ways to handle collisions (open addressing, etc.). We will not discuss them. Probably the results are of the same nature for all ways.

Denote by $\Delta(n, T)$ the set of all the dictionaries which contain T binary n-length words, n and T are natural. A map $f: E^n \to [0, m-1]$ is said to be uniform, if $|f^{-1}(0)| = \cdots = |f^{-1}(m-1)|$.

Turn to the question we are concerned with: how to choose a good hash-function? Suppose any dictionary of $\Delta(n, T)$ is equally likely to appear. Then, as it is shown in Claim 2, the mathematical expectation of the index $I(f, D)$ over $D \in \Delta(n, T)$ is minimal iff f is uniform. That is why a nonuniform map will not be used as a hash-function. But which uniform map to choose? Each one of them is on average just as good as any other. On the other hand, whichever map f one selects, there is a dictionary D on which f performs terribly, i.e., $I(f, D)$ is very large. So it looks more promising to seek a good set of hash functions rather than a single one. Such a set should for any directionary contain a good enough uniform hash-function. By the way, many authors propose usually not one, but several hash-functions. A chosen function being bad they recommend to rehash, i.e., to drop it and to take another. We came to the following

DEFINITION. Given a wordlength n, a cardinality of dictionaries T, a loading factor α and a desirable index level a, a set M of uniform maps from E^n to $[0, m-1]$, $m = T/\alpha$, is called a-hash-set, if for any $D \in \Delta(n, T)$ there is a map $f \in M$ such that $I(f, D) \leqslant a$. The symbol $N(n, T, \alpha, a)$ stands for the cardinality of the minimal a-hash-set.

If one wants to get a hash-function whose index on a given disctionary is

less than a number a, one might pick a function f at random out of an a-hash-set and repeat the procedure if not in luck, i.e., if $I(f, D) > a$. The less the cardinality of the a-hash-set, the more the probability to make a good choice and the less the number of rehashing.

Thus, it is of interest to find $N(n, T, \alpha, a)$ and to develop nearly minimal hash-sets. It is the first aim of the paper.

The same question: how to choose a good hash-function? may be understood another way. It is necessary to choose the best function f such that $I(f, D) \leqslant a$, given a dictionary D and a performance level a. Functions are to be calculated by a computer, so we may decide that the best function is one with either the shortest program or the minimal time of calculation or the minimal computer space used. Albeit any of those decisions is possible, we select the first one, so that the complexity $L(f)$ of a function f is meant to be the bit length of a shortest program calculating f, i.e., $L(f)$ is Kolmogorov complexity of f. The reason to concentrate primarily on program length is that search methods differ much more in this complexity measure than in two others. Still, some attention will be paid to running time and computer space as well.

We will say some more about program length. Being fed first with a program and then with a word x an initially empty computer produces the hash-address $f(x)$. If one wishes one can distinguish two parts of a program. The first one is input information and is a sort of compressed representation of a dictionary D. The second or controlling part of the program consists of several computer instructions which say how to operate both the input information and a word x to produce $f(x)$. The controlling part does not depend on a dictionary. The main contribution to the program length is made by the input information. The program length is computer independent asymptotically. The diagram of address computing is shown in Fig. 1.

Given a dictionary D, a loading factor α and a performance level a, let $L(D, \alpha, a)$ stand for the minimum program length $L(f)$ of all uniform maps f from E^n to $[0, m-1]$, $m = |D|/\alpha$ with $I(f, D) \leqslant a$, and let $L(n, T, \alpha, a) = \sup_{D \in \Delta(n,T)} L(D, \alpha, a)$. The number $L(n, T, \alpha, a)$ is the bit length of a shortest program producing hash-addresses for the worst dictionary $D \in \Delta(n, T)$ with the index not exceeding a. The second aim of the paper is to provide any $D \in \Delta(n, T)$ with a uniform map f such that $I(f, D) \leqslant a$ and f is simple enough. The cardinality T is presumed to be a divisor of 2^n without loss of generality.

There is a tradeoff between the desired performance level α and the complexity $L(n, T, \alpha, a)$: the more the level a, the less the complexity. We are interested in tracing this tradeoff.

Such a statement of the hashing problem seems natural enough, though, probably it was not discussed earlier (see Knuth, 1973; Knott, 1975) and the references there.

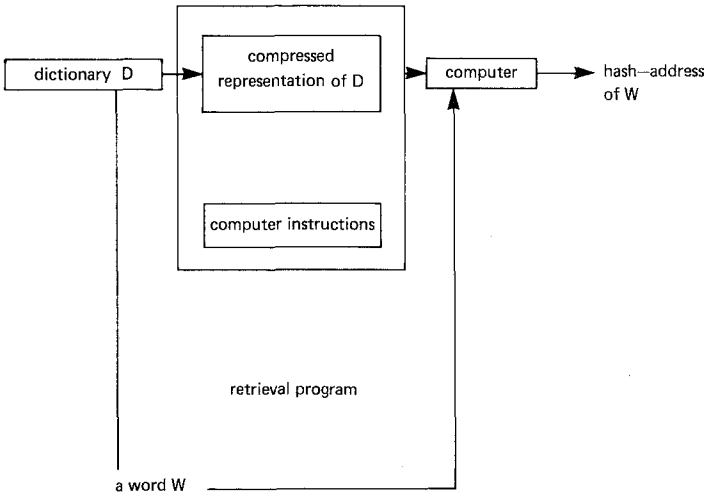

FIG. 1. Hash-address computation.

Albeit we will busy ourselves with the set $\Delta(n, T)$, other sets may be studied the same way. Our approach to the problem enables us to discuss both collision-free and hash-search methods on equal footing. Next we review the main of them and give our results on such a background.

1.2. *A Review of Search Methods and the Main Results of the Paper*

Basic computational characteristics of a key-address map are its program length, running time, and space used. They are displayed in Table I for main retrieval algorithms.

Being stored with a program, a computer produces the address $f(x)$ from a word x. We measure in bits the sizes of both the program and the additional memory necessary for calculations. Although program size does not depend asymptotically on the computer, space size and running time do. Our computer is supposed to have random access memory and all usual operations. When estimating the running time, the length of its machine word is supposed to equal the length n of words of a dictionary.

A map from E^n to $[0, m-1]$, $m = T/\alpha$, is called strong for a dictionary D, if it takes all words out of D to a number the same for all $x \in E^n \backslash D$, and different from all $f(x)$, $x \in D$, so that one can tell through $f(x)$ whether the search is successful or not, i.e., whether $x \in D$ or $x \notin D$. An arbitrary map is called weak, so any strong map is weak as well. That is why the lower bound of the program length for weak algorithms is much less than that for strong algorithms. Those bounds are $L(n, T, \alpha, 0) \geqslant$

$T \log_2 e(1 + (1/\alpha)(1 - \alpha) \ln(1 - \alpha))$ and $L^s(n, T, \alpha, 0) \geqslant T(n - \log_2 T)$, respectively, where $L^s(n, T, \alpha, 0)$ is the program length of the best collision-free strong map for the worst dictionary $D \in \Delta(n, T)$, $\log x = \log_2 x$. The first bound was proved in Krichevsky, (1978a), the second one in Krichevsky, (1978b), under the condition $0 < \lim(\log T/n) < 1$.

In this paper the scope of lower bounds is enlarged to cover the hashing: $L(n, T, \alpha, a) > cT$ if $a < \alpha$; $L(n, T, \alpha, \alpha) > c \log T$; $L(n, T, \alpha, a) > \log(n - 2 \log T)$, if $a > \alpha$. (Throughout the paper c is a positive constant, and the same letter c may stand for different constants). Conditions for those bounds to hold are in the theorem.

The point is that the lower bound for hashing program length $L(n, T, \alpha, a)$ nearly equals one for collision-free methods as long as we want to have only maps with the index less that the loading factor α. Thus, there is not much use in hashing until $a < \alpha$. However, no sooner are we ready to accept maps whose index equals α, as the lower bound will decrease quite essentially so that the program length of hashing becomes much less than one of collision-free algorithms. Our bounds are tight enough. It is interesting to compare the program length of a search algorithm with those bounds.

Proceed to Table I. The first part of the unordered search program for a dictionary D is the concatenation of T n-length words w, $w \in D$. The instructions of the second part say: compare the first n-length subword of the concatenation with an n-length word x. If they are equal, then $f(x) = 0$, else go to the second subword, and so on. If $x \notin D$, $f(x) = \varnothing$, thus the method is strong. The running time equals cT in the worst case.

The first part of the logarithmic search program for D is the concatenation of words of D lexicographically ordered. The instructions say: compare x with the middle n-length subword of the concatenation and then go to the left or right, etc. The program length is the same as with unordered search, whereas the running time is $O(\log T)$.

The enumerative algorithm is as follows. Let val w be the number whose binary notation is a word w. Associate to a dictionary D the binary 2^n-length word $\chi(D)$, whose val wth letter is either 1, if $w \in D$, or 0, if $w \notin D$. The word $\chi(D)$ contains $T = |D|$ units and $2^n - T$ zeros. Index all binary 2^n-length words with T units from 0 to $C_{2^n}^T - 1$. Make the index of the word $\chi(D)$ to be the first part of the enumerative program. Its length equals $\log C_{2^n}^T = T(n - \log T)$ bits asymptotically, if $0 < \lim(\log T/n) < 1$, as it is easily seen from Stirling formula. To find the address $f(x)$ of a word x first write $\chi(D)$ via its index, i.e., the first part of the program. The algorithm from Cover (1973) can be used for this purpose. The running time and additional space are enormous: $O(2^n)$. Then count the number of units to the left of the val xth position of $\chi(D)$. It is just $f(x)$. Thus, enumerative algorithm meets the lower bound of program length, but it consumes a lot of time and space.

Digital search in Knuth (1973) is defined by a binary tree with T leaves. All $2T-1$ nodes of the tree are numbered, and each of them is supplied with the numbers of its sons. These numbers are called links. The links of the leaves are $\varnothing$. The first part of the digital search program is the concatenation of $2T-1$ $\lceil \log 2T \rceil$-length links. The second part says: take the first digit of a word x and go to either the left son of the root if it is 0, or else to the right one, and so on. If all words of the dictionary have the same first digit, it is no use to look at it, and the algorithm is to be slightly modified. So, the program length is $CT \log T$, the running time is $O(n)$. The algorithm is weak.

The program length of the digital search can be lessened up to $CT \log n$. For this purpose one should examine the digits of a word x in a special succession, see Krichevsky (1978b) rather than in their natural order as in Knuth (1973). There is a strong variant of the same algorithm in that very paper. It meets the lower bound $T(n - \log T)$ for the program length of the strong search.

A weak retrieval algorithm with the shortest program was developed in Krichevsky (1978a). Its program length meets the aforesaid lower bound which equals $T \log_2 e$, if $\alpha = 1$.

Note that "Oxford Dictionary" by A. S. Hornby gives such a definition: "a dictionary is a book dealing with the words of a language, and arranged in ABC order." But both weak and strong optimal retrieval algorithms Krichevsky (1978a, 1978b) give up the ABC arrangement; to achieve the shortest program length, he arranges the words in orders quite different from alphabetical.

The review of collision-free algorithms is concluded. Go to hashing. If one wants to get a hash-function with index $a < \alpha$, one can take the weak map of Krichevsky (1978a). Its program length equals the lower bound to within a factor C. If index $a = \alpha$, we develop in the theorem for a dictionary D a map with program complexity $n(1 + o(1))$ which equals the lower bound $C \log T$ to within a factor C. The map is called K-linear and is as follows: Let $m = 2^\mu$, μ is natural and a divisor of n. Choose an irreducible polynomial $g(x)$ of degree μ with coefficients from $GF(2)$. Any polynomial over $GF(2)$ of $\mu - 1$ or less degree can be considered a residue modulo $g(x)$ and an element of $GF(2^\mu)$. Any binary n-length vector b defines a linear map φ_b. Partition n-length vectors b and x into n/μ successive μ-length subvectors $b_1, \ldots, b_{n/\mu}$ and $x_1, \ldots, x_{n/\mu}$. Let the hash-address of a word x be $\varphi_b(x) = x_1 b_1 + \cdots + x_{n/\mu} \cdot b_{n/\mu}$, let x_i and b_i be members of $GF(2^\mu)$, and multiplication and addition be performed in this field. The hash-address $\varphi_b(x)$ is a μ-length vector representing a number from the range $[0, m-1]$. The n-length vector b is the first part of the hashing program, the controlling part is a set of computer instructions to calculate $\varphi_b(x)$. As it is proved in the theorem, for any dictionary D there is a vector b such that $I(\varphi_b, D) \leqslant \alpha$.

If we accept maps whose index exceeds the loading factor α, then the

following K-polynomial maps are of use. Such a map φ_C is defined by a μ-length vector C:

$$\varphi_C(x) = \sum_{i=1}^{n/\mu} x_i \cdot C^{i-1}.$$

The calculations are within $GF(2^\mu)$, the hash-address is within $[0, m-1]$. The vector C is the first part of the hashing program. As it is proved in the theorem, for any dictionary D, there is a vector C such that $I(\varphi_C, D) \leqslant \alpha n/\log m(1+o(1))$. Program length of K-polynomial hashing agrees with the lower bound less satisfactorily than that of K-linear hashing.

Hash-sets which a hash-function can be taken at random from are also developed in the theorem. The lower bound for $\log N(n, T, \alpha, a)$ is the same as for $L(n, T, \alpha, a)$. The upper bound for the first quantity is slightly better than for the second one: $\log N(n, T, \alpha, a) < \log T$, $a = \alpha$; and $\log N(n, T, \alpha, a) < \frac{1}{2} \log T$, $a > \alpha$. The reason for that is that hash-sets are constructed implicitly, using "covering lemma," see Section 9. This lemma provides a small enough hash-set M, but it cannot guarantee that functions $f \in M$ have short programs.

It seems appropriate to mention here an interesting paper of Jaeschke (1981). He shows that for any distionary D there are constants C, A, r such that the map $x \to \lfloor C/Ax + r \rfloor$ is injective on D. The index of the map equals 0, the loading factor $\alpha = 1$. Such a map is called perfect. It is said in the paper cited that the constants C, A, r are regretfully very great. An explanation of this phenomenon is provided by the lower bound of $L(n, T, \alpha, 0)$ which is $T \log e$. The constants C, A, r play the role of the first part of a computer program for perfect hashing. Thus, the sum of their logarithms could never be less than $T \log e$, which is a big enough number, because $\log T$ usually equals Cn.

We end our review with an application of hashing to the information theory. Let a source produce a sequence of n-length words from a dictionary D, $|D| = T$. The sequence is to be transmitted through a noiseless channel. Choose a loading factor α and a hash-function f, $I(f, D) = \alpha$, program length $L(f)$ of f is $C \log T$. Such a function exists thanks to the theorem. Encode a word x, $x \in$, by the concatenation of $f(x)$ and of the number of x in the succession of words y, colliding with x, i.e., $f(y) = f(x)$, $y \in D$. The average codelength equals $\lceil \log(T/\alpha) \rceil + (1/T) \sum_{k=0}^{m-1} i_k \lceil \log i_k \rceil$, where $m = T/\alpha$, $i = (i_0, \ldots, i_{m-1})$ is the signature of f on D. Obviously, this codelength is not more than $\log T - \log \alpha + I(f, D) + 1 \leqslant \log T + \alpha - \log \alpha + 2 = \log T + C$, see (1.2). So, the average number of bits transmitted is slightly more than the minimum value $\log T$. But the program lengths of encoding and decoding functions are $O(\log T) = O(n)$. The average time to find x in the list of all y, $f(y) = f(x)$, is $t(f, D) = \alpha/2 + 1 = C$. So, the program and time perfor-

mances of such a hashing transmiting scheme are very good, much better, than those of any collision-free encoding procedure. A disadvantage of the scheme is that the separate lists of colliding words are to be kept on the input and output. One hash-function may suit several disctionaries.

1.3. *The Structure of the Paper*

The main result is the theorem (Sect. 6) which yields asymptotic bounds for $L(n, T, \alpha, a)$ and $N(n, T, \alpha, a)$. Its proof is rather longish and is divided into three paragraphs: Section 7—lower bounds, Section 8—implicit upper bound for the cardinality of hash-sets and Section 9—explicit upper bound for both that cardinality and program complexity. Three cases: (i) the index is less than the loading factor, (ii) they are equal, (iii) the index is more than the loading factor, are to be considered separately. Paragraphs 2–5 are preliminary.

The upper bounds of program length $L(n, T, \alpha, a)$ are obtained in Section 9 via the general method of hash-sets construction from Section 2. Perhaps, some other interesting hash-functions, apart from K-linear and K-polynomial, may be produced by the method.

To get other bounds, we find (Sect. 3) for how many dictionaries an arbitrary function has an index a and how many functions f are there for a dictionary D, such that $I(f, D) \leqslant a$. Both those numbers are approximated by Poisson distribution. For the approximation to hold the condition $\lim T^2/2^n = 0$ is required. But the results, probably, remains the same even without this condition. In Section 5, Lemma 3 we find asymptotic bounds for the number of dictionaries on which f has index either less or more than the given number a. These bounds are obtained by means of "large deviations theorems" for sums of independent variables of probability theory. We use the inequalities of Nagaev, Petrov and Bernstein. If, on the one hand, we know the number of all dictionaries in $\Delta(n, T)$ and, on the other hand, the number of dictionaries D, for which $I(f, D) \leqslant a$, then their quotient is a lower bound for $N(n, T, \alpha, a)$. This is the way to obtain lower bounds for the cardinalities of a-hash-sets, $a < \alpha$, Section 8(i). But if $a = \alpha$, then the index of a function is less than α on a half of all the dictionaries in $\Delta(n, T)$. Hence, the aforesaid way of reasoning produces a trivial bound $N(n, T, \alpha, a) \geqslant 2$. To receive a tighter bound $\log N(n, T, \alpha, \alpha) \geqslant C \log T$ of the theorem, it is necessary to use a less obvious method. Given an a-hash-set $F = \{f_0, \ldots, f_{|F|-1}\}$, we restrict the maps f_i to a subset $X \subseteq E^n$ in order to decrease their ranges and to make the mathematical expectations $EI(f_i, D)$ more than a, $D \subseteq X$, $i = 0, \ldots, |F| - 1$. The index of a map f_i will be less than a only for a small part of dictionaries $D \subseteq X$, and hence $|F|$ must be large enough. To implement the idea we use Chebyshev inequality (Lemma 5) and an inequality between $\sum_{i=0}^{M-1} \lambda_i^3$ and $\sum_{i=0}^{M-1} \lambda_i^2$, Lemma 4.

The maps f_i, being uniform on E^n, cease to be such on X. That is why we

could not use the inequalities of Lemma 3 and have to handle dependent not identically distributed variables in Lemma 5.

The lower bounds for $L(n, T, \alpha, a)$ are deduced from those for $N(n, T, \alpha, a)$ through the inequality

$$L(n, T, \alpha, a) \geqslant \log N(n, T, \alpha, a) - 1. \tag{1.4}$$

The inequality holds, because there are no more than 2^{L+1} functions whose program length does not exceed L, $L > 0$. If (1.4) is not true, then $N(n, T, \alpha, a) \leqslant 2^{L+1} < 2^{\log N(n,T,\alpha,a)-1+1}$—a contradiction.

In Section 4 we prove that the best single hash-function for $\Delta(n, T)$ is uniform.

In Section 8 implicit upper bounds for $N(n, T, \alpha, a)$ are obtained via Nechiporuk's "covering lemma," an equivalent of random coding.

2. A Method to Construct Hash-Sets

Let n and m be natural, $\Phi = \{\varphi_1, ..., \varphi_{|\Phi|}\}$ be a set of maps from E^n to $\{0, ..., m-1\}$, $|\Phi|$ be the cardinality of Φ. Tensor product $\otimes\Phi$ of $\varphi_1, ..., \varphi_{|\Phi|}$ is the map which takes a word $x \in E^n$ to the concatenation $\varphi_1(x), ..., \varphi_{|\Phi|}(x)$. The distance $\rho(a, b)$ between two equal length words a and b in the alphabet $\{0, ..., m-1\}$ is understood to be Hamming, i.e., it is the number of positions which have different letters.

Claim 1. Let $\Phi = \{\varphi_1, ..., \varphi_{|\Phi|}\}$ be a set of maps from E^n to $\{0, 1, ..., m-1\}$, $n > 0$, $m > 0$, such that $\min_{x,y \in E^n} \rho(\otimes\Phi x, \otimes\Phi y) = r$, $r > 0$. Then for any dictionary D containing n-length binary words there is a map $\varphi_0 \in \Phi$ whose index on D does not exceed

$$I(\varphi_0, D) \leqslant (|D| - 1)\left(1 - \frac{r}{|\Phi|}\right).$$

Proof. Make a table for a dictionary D. The lines of the table correspond to the words of D, the columns, to the maps of Φ. We put $\varphi(w)$ at the intersection point of the w-line and φ-column, $w \in D$, $\varphi \in \Phi$. Sum up all the distances between the lines of the table. On the one hand, the sum is not less that $(|D|(|D|-1)/2)r$, from the conditions of the lemma. On the other hand, the sum may be obtained by summing $|\Phi|$ addends up. The ith of them equals the sum of all the distances between letters of the ith column, $i = 1, ..., |\Phi|$. Hence, there is a map $\varphi_0 \in \Phi$ such that the corresponding addend is not less than $(|D|(|D|-1)/2|\Phi|) \cdot r$. Let i_k be the number of words w, $w \in D$, for which $\varphi_0(w) = k$, $k \in \{0, 1, ..., m-1\}$.

Obviously,

$$\sum_{k=0}^{m-1} i_k = |D|. \tag{2.1}$$

The addend corresponding to φ_0 equals $\sum_{w_1, w_2 \in D} \rho(\varphi_0(w_1), \varphi_0(w_2)) = |D|(|D|-1)/2 - \sum_{k=0}^{m-1} C_{i_k}^2$, because $\rho(\varphi_0(w_1), \varphi_0(w_2)) = 0$, if $\varphi_0(w_1) = \varphi_0(w_2)$. Thus,

$$\frac{|D|\,(|D|-1)}{2} - \sum_{k=0}^{m-1} C_{i_k}^2 \geqslant \frac{|D|\,(|D|-1)}{2\,|\Phi|} \cdot r. \tag{2.2}$$

According to (1.2),

$$I(\varphi_0, D) = \frac{1}{|D|} \cdot \sum_{k=0}^{m-1} i_k^2 - 1. \tag{2.3}$$

From (2.1), (2.2), and (2.3) we get

$$I(\varphi_0, D) \leqslant (|D|-1)\left(1 - \frac{r}{|\Phi|}\right). \qquad \text{Q.E.D.}$$

3. Cluster Distribution

We will use multiindex notation. If $a = (a_1, \ldots, a_s)$, $b = (b_1, \ldots, b_s)$ are vectors, then $a! = a_1! \cdot \cdots \cdot a_s!$, $a^b = a_1^{b_1} \cdot \cdots \cdot a_s^{b_s}$, $|a| = a_1 + \cdots + a_s$.

Let there be a set $X \subseteq E^n$, a map $f: X \to [0, M-1]$, and a dictionary D, $|D| = T$, $T > 0$. The signature of a uniform map is denoted by the letter u, so that $u = (|X|/M, \ldots, |X|/M)$.

The symbol $p_x(i)$ stands for the probability to meet a dictionary on which a map with the signature x on X has got the signature i, provided all dictionaries D, $D \subseteq X$, $|D| = |i| = T$, have the same probability. That probability depends only on the signature x of a map. The symbol $\bar{p}_x(i)$ stands for the probability to meet a map having the signatures x on X and i on a dictionary D, provided all maps f with the signature x have the same probability.

Lemma 1. *For a set X, a number T and vectors x and i, $|x| = |X|$, $|i| = T$, the probabilities $p_x(i)$ and $\bar{p}_x(i)$ are equal and*

$$p_x(i) = \bar{p}_x(i) = \frac{x!\,|i|!\,(|x|-|i|)!}{i!\,(x-i)!\,|x|!}.$$

Proof. Let f be a map with a signature x. To obtain a dictionary $D \subseteq X$ on which f has a signature i one shall select i_k words from x_k words taken

by f to k, $k = 0,..., M-1$. Therefore, the number of such dictionaries equals $C_{x_0}^{i_0} \cdot \cdots \cdot C_{x_{M-1}}^{i_{M-1}} = x!/i!(x-i)!$. Dividing that by the number $C_{|X|}^{T}$ of all dictionaries we obtain p_x^i.

A map f with the signatures x on X and i on a dictionary D takes i_k words of D and $x_k - i_k$ words of $X \backslash D$ to k, $k = 0,..., M-1$. There are $T!/i!$ ways to partition D into M subsets with cardinalities $(i_0,..., i_{M-1}) = i$. Likewise, there are $(|X| - T)!/(x-i)!$ ways to partition $X \backslash D$ into M subsets with cardinalities $x - i = (x_0 - i_0,..., x_{M-1} - i_{M-1})$. Multiplying those numbers and dividing the product by the number $|x|!/x!$ of all maps with signature x, we obtain that $p_x(i) = \bar{p}_x(i)$. Q.E.D.

LEMMA 2. *Let X be a set, T be a number, x and i be M-dimensional vectors, $M \geqslant 1$, $\lambda = (T/|x|)x$, $T^2/|X| \to 0$. Then the distribution $p_x(i)$ is majorized by Poisson distribution to within the factor $T!\, e^T T^{-T}$:*

$$p_x(i) \leqslant \frac{\lambda^i}{i!} e^{-T}(T!\, e^T T^{-T}).$$

Moreover, if x is uniform, i.e., $x = (|X|/M,..., |X|/M)$, and the condition $(1/T)\sum_{k=0}^{M-1} i_k^2 \leqslant C$ holds, then

$$p_u(i) = \frac{\lambda^i}{i!} e^{-T}(1 + o(1))(T!\, e^T T^{-T}),$$

$o(1)$ is uniform over i, $T!\, e^T T^{-T} \sim \sqrt{2\pi T}$.

Proof. From Lemma 1 we have

$$p_x(i) = \frac{T!}{i!} \frac{x_0 \cdots (x_0 - i_0 + 1) \cdots (x_{M-1} - i_{M-1} + 1)}{|x|\,(|X| - 1) \cdots (|X| - T + 1)}. \tag{3.1}$$

The equality (3.1) implies

$$p_x(i) \leqslant \frac{T!}{i!} \left(\frac{x_0}{|X|}\right)^{i_0} \cdots \left(\frac{x_{M-1}}{|X|}\right)^{i_{M-1}} \cdot \gamma_1, \tag{3.2}$$

where

$$\gamma_1 = \left(\left(1 - \frac{1}{|X|}\right) \cdots \left(1 - \frac{T-1}{|X|}\right)\right)^{-1}.$$

If $x_0 = \cdots = x_{M-1} = |X|/T$, then

$$p_u(i) = \frac{T!}{i!} \left(\frac{x_0}{|X|}\right)^{i_0} \cdots \left(\frac{x_{M-1}}{|X|}\right)^{i_{M-1}} \gamma_2 \gamma_1, \tag{3.3}$$

where

$$\gamma_2 = \left(1 - \frac{T}{|X|}\right) \cdots \left(1 - \frac{i_0 - 1}{|X|} \cdot T\right) \cdots \left(1 - \frac{T}{|X|}\right) \cdots \left(1 - \frac{i_{M-1} - 1}{|X|} T\right).$$

As it is easily seen, the inequality

$$-x \geqslant \ln(1 - x) \geqslant -2x$$

holds, $0 \leqslant x \leqslant \frac{1}{2}$. Using it, we obtain

$$0 \leqslant \ln \gamma_1 \leqslant \frac{2}{|X|} \cdot \sum_{j=0}^{T-1} j = \frac{T(T-1)}{|X|}. \tag{3.4}$$

The same inequality yields

$$|\ln \gamma_2| \leqslant C \cdot \frac{T}{|X|} \sum_{k=0}^{M-1} i_k^2. \tag{3.5}$$

The inequality (3.4) and the condition $T^2/|X| \to 0$ yield

$$\gamma_1 \to 1. \tag{3.6}$$

The same condition (3.5) and the inequality $\sum_{k=0}^{M-1} i_k^2 \leqslant CT$ imply

$$\gamma_2 \to 1. \tag{3.7}$$

Formulae (3.2) and (3.6) imply the first claim of the lemma, whereas (3.3), (3.6), and (3.7)—the second one. The asymptotic equality $T!\, e^T \cdot T^{-T} \sim \sqrt{2\pi T}$ is equivalent to the Stirling formula. Q.E.D.

4. Single Hash-Function

Claim 2. The mathematical expectation $EI(f, D)$ of the index of a map f over all $D \subseteq X$ equals

$$\frac{T-1}{|X|\,(|X|-1)} \left(\sum_{k=0}^{M-1} x_k^2 - 1\right),$$

where $x = (x_0, \ldots, x_{M-1})$ is the signature of f on X. The mathematical expectation is minimal iff f is uniform. The minimum tends to $\beta = T/M$, as $T/|X| \to 0$.

Proof. Let $y = (y_0, \ldots, y_{M-1})$, $\overline{1} = (1, \ldots, 1)$, $p(y) = (\overline{1} + y)^x \cdot 1/C_{|X|}^T$. The

probability p_x^i equals the coefficient at y^i in the polynomial $p(y)$. The mathematical expectation Ei_k^2, $k = 0,..., M-1$ equals the coefficient at z^T in

$$y_k \frac{\partial}{\partial y_k} y_k \frac{\partial}{\partial y_k} p(y)|_{y_0 = y_1 = \cdots = y_{M-1} = z}.$$

Differentiating, we obtain

$$Ei_k^2 = \frac{T \cdot x_k}{|X|} \cdot \left(\frac{(x_k - 1)(T-1)}{|X| - 1} + 1 \right).$$

Substituting it into (1.2), we find $EI(f, D)$. The minimum of $\sum_{k=0}^{M-1} x_k^2$, under the condition $\sum_{k=0}^{M-1} x_k = |X|$, is assumed at the point $x_0 = \cdots = x_{M-1} = |X|/M$. Finally, we easily find $\lim \min_{T/|X| \to 0} I(f, D) = T/M = \beta$. Q.E.D.

5. Large Deviations Probabilities

Given a set X, numbers T, M and a vector $x = (x_0,..., x_{M-1})$, we denote by $p_x(I \gtrsim b)$ the probability to meet a dictionary $D \subseteq X$, $|D| = T$ on which a function f with the signature x has the index $I(f, D) \gtrsim b$, $b > 0$:

$$\begin{aligned} p_x(I \gtrsim b) &= \sum p_x(i) \\ \sum_{k=0}^{M-1} i_k^2 &\gtrsim bT + T, \qquad |i| = T. \end{aligned} \tag{5.1}$$

That probability is the same for all functions with the signature x.

Lemma 3. *Let $M, T, |X|$ tend to infinity, $T/M = \beta = \text{const}$, $T^2/|X| \to 0$. Then for any $C > 0$ there are positive constants C_1, C_2 such that*

(i) $p_u(I \leqslant \beta - C) < e^{-C_1 T}$

(ii) $p_u(I > \beta + C) < e^{-C_2 \sqrt{T} \ln T}$.

Proof. Let $\lambda = (T/|X|)u = (\beta,..., \beta)$. From Lemma 2 and (5.1) we have

$$\begin{aligned} p_u(I \leqslant \beta - C) &= e^{-T} \sum \frac{\lambda^i}{i!} \\ |i| = T, \qquad \sum_{k=0}^{M-1} i_k^2 &\leqslant \beta T + T - CT. \end{aligned} \tag{5.2}$$

Ignoring the condition $|i| = T$ may only increase the right-hand side of (5.2):

$$p_u(I \leqslant \beta - C) \leqslant e^{-T} \sum \frac{\lambda^i}{i!} \sqrt{2\pi T}\,(1 + o(1)) \qquad \sum_{k=0}^{M-1} i_k^2 \leqslant \beta T + T - CT. \tag{5.3}$$

Consider a sequence of independent identically distributed stochastic variables $\xi_0, \ldots, \xi_{M-1}$ with the distribution

$$p(\xi_j = r^2) = e^{-\beta} \cdot \frac{\beta^r}{r!}, \qquad r \geqslant 0,\ 0 \leqslant j \leqslant M-1. \tag{5.4}$$

Their first two moments are

$$E\xi_j = \beta + \beta^2, \qquad E(\xi_j - E\xi_j)^2 = 4\beta^3 + 6\beta^2 + \beta. \tag{5.5}$$

From (5.4) and (5.5),

$$p\left(\sum_{j=0}^{M-1} \xi_j - E\xi_j \leqslant -CT\right) = e^{-T} \sum \frac{\lambda^i}{i!} \qquad \sum_{k=0}^{M-1} i_k^2 \leqslant \beta T + T - CT. \tag{5.6}$$

(i) Deviations to the left. Consider a function $\varphi(t)$,

$$\varphi(t) = e^{-t(\xi_j - E\xi_j)} = e^{-\beta} \sum_{r=0}^{\infty} e^{-t(r^2 - E\xi_j)} \frac{\beta^r}{r!}, \qquad j = 0, \ldots, M-1. \tag{5.7}$$

Series (5.7) converges uniformly, $0 \leqslant t \leqslant 1$. Obviously,

$$\varphi(0) = 1, \varphi'(0) = 0, \varphi''(t) = e^{-\beta} \sum_{r=0}^{\infty} e^{-t(r^2 - E\xi_j)} (r^2 - E\xi_j)^2 \frac{\beta^r}{r!} \leqslant e^{tE\xi_j} \cdot E(\xi_j - E\xi_j)^2 \leqslant C',\ 0 \leqslant t \leqslant 1. \tag{5.8}$$

Use (5.8) in Taylor expansion for $\varphi(t)$:

$$\varphi(t) \leqslant 1 + \frac{t^2}{2} \cdot C'. \tag{5.9}$$

Inequality

$$Ee^{-t(\xi_j - E\xi_j)} \leqslant e^{C'/2 \cdot t^2}, \qquad 0 \leqslant t \leqslant 1$$

follows (5.7), (5.9) and the obvious inequality $e^{C' \cdot t^2/2} \geqslant 1 + C' \cdot t^2/2$. Now we can use the inequalities from Bernstein (1946) and from Petrov (1972, p. 70, Theorem 15), yielding

$$p\left(\sum_{j=0}^{M-1} (\xi_j - E\xi_j) < -CT\right) \leqslant e^{-C^2 \cdot T^2/2MC'} = e^{-C'' \cdot T} \qquad \text{if} \quad CT < MC',$$

$$\leqslant e^{-CT/2} \qquad \text{if} \quad CT \geqslant MC'.$$

Letting $C_1 = \min(C'', C/2)$ we obtain from here, (5.6) and (5.3), Claim (i).

(ii) Deviations to the right. Here we use Theorem 2.3 from Nagaev (1979, p. 765). Let $g(x) = C\sqrt{x} \ln x$. The generalized g-moment b_g of the variable ξ_j equals

$$b_g = e^{-\beta} \sum_{k=0}^{\infty} e^{C\sqrt{k^2 - E\xi_j} \ln(k^2 - E\xi_j)} \cdot \frac{\beta^k}{k!}$$

$$\leqslant e^{-\beta} \sum_{k=0}^{\infty} e^{(C_1 k \ln k - k \ln k)(1 + o(1))}.$$

This moment exists for small enough C, $j = 0, \ldots, M-1$.

Taylor expansion for the exponent in Lagrange form and Stirling formula yield

$$p_u((\xi_j - E\xi_j) > CT) = e^{-\beta} \sum_{k^2 > E\xi_j + CT} \frac{\beta^k}{k!} \leqslant e^{-\beta} \cdot e^{\beta} \cdot \frac{\beta^{\sqrt{CT + E\xi_j}}}{(\sqrt{CT + E\xi_j})!}$$

$$\leqslant e^{-C\sqrt{T} \ln T}.$$

This last inequality and the existence of the g-moment suffice for the Nagaev inequality to hold. It yields

$$p_u\left(\sum_{j=0}^{M-1} \xi_j - E\xi_j > CT\right) \leqslant e^{-C_1\sqrt{T} \ln T}.$$

That inequality along with the one analogous to the (5.3) inequality

$$p_u(I > \beta + C) \leqslant e^{-T} \cdot \sum \frac{\lambda^i}{i!} (1 + o(1)) \sqrt{2\pi T}$$

$$\sum_{k=0}^{M-1} i_k^2 \geqslant \beta T + T + CT,$$

and analogous to the (5.6) equality

$$p_u\left(\sum_{j=0}^{M-1}(\xi_j - E\xi_j) > CT\right) = e^{-T}\sum \frac{\lambda^i}{i!}$$

$$\sum_{k=0}^{M-1} i_k^2 \geqslant \beta T + T + CT$$

yield Claim (ii) of the lemma. Q.E.D.

In Lemma 5 we need

LEMMA 4. *Let there be $M > 0$ numbers $\lambda_0,\ldots,\lambda_{M-1}$, $\lambda_0 + \cdots + \lambda_{M-1} = T$, $T > 0$, $\beta = T/M$. Then*

$$\sum_{i=0}^{M-1} \lambda_i^3 \leqslant \left(\sum_{i=0}^{M-1} \lambda_i^2 - M\beta^2\right)^{3/2} + 3\beta\left(\sum_{i=0}^{M-1} \lambda_i^2 - M\beta^2\right) + M\beta^3.$$

Proof. Find the maximum of the function $\sigma_3 = \sum_{i=0}^{M-1} \lambda_i^3$ under the conditions

$$\sigma_1 = \lambda_0 + \cdots + \lambda_{M-1} = T, \qquad \sigma_2 = \sum_{i=0}^{M-1} \lambda_i^2 = \text{const.}$$

Lagrange function is $\sigma_3 - \gamma_1\sigma_1 - \gamma_2\sigma_2$, its partial derivatives are $3\lambda_i^2 - 2\gamma_1\lambda_i - \gamma_2 = 0$, $i = 0,\ldots, M-1$. We have a second degree equation. Hence, the variables $\lambda_0,\ldots,\lambda_{M-1}$ can take no more than two values at extremal points. Denote those values by $x+\beta$ and $y+\beta$, and let the first of them be taken s times, the second one—$M-s$ times, $0 \leqslant s \leqslant M$. The conditions take shape

$$sx + (M-s)\,y = 0 \tag{5.10}$$

and

$$sx^2 + (M-s)\,y^2 + M\beta^2 = \sigma_2 = \text{const.} \tag{5.11}$$

For the function σ_3 we have

$$\sigma_3 = sx^3 + (M-s)\,y^3 + 3\beta(sx^2 + (M-s)\,y^2) + M\beta^3. \tag{5.12}$$

If either $s = 0$ or $s = M$, then the inequality of Lemma 3 turns to an equality. If $0 < s < M$, then from (5.10) and (5.11)

$$y = -\frac{s}{M-s}x, \qquad x = \pm\sqrt{(M-s)/Ms}\,\sqrt{\sigma_2 - M\beta^2}. \tag{5.13}$$

Substitute (5.13) into (5.12):

$$\sigma_3 = \pm(\sqrt{\sigma_2 - M\beta^2})^3 \cdot \frac{M-2s}{\sqrt{Ms(M-s)}} + 3\beta(\sigma_2 - M\beta^2) + M\beta^3. \quad (5.14)$$

The function $(M-2s)/\sqrt{s(M-s)}$ is symmetric with respect to $s = M/2$ and is decreasing, $1 \leqslant s \leqslant M-1$. Hence,

$$\frac{M-2s}{\sqrt{Ms(M-s)}} \leqslant \frac{M-2}{\sqrt{M(M-1)}}. \quad (5.15)$$

As it is easily seen, if $M \geqslant 2$,

$$\frac{M-2}{\sqrt{M(M-1)}} \leqslant 1. \quad (5.16)$$

Equality (5.14) holds at the extremal points. Using (5.15) and (5.16), we obtain the inequality of the lemma for $M \geqslant 2$. If $M = 1$ that inequality holds obviously. Q.E.D.

LEMMA 5. *Let X be a set, M, T numbers, x an M-dimensional vector; $T^2/|X| \to 0$, $M, T, |X|$ tend to infinity, $\alpha = \text{const.}$, $CT^{1/2} \geqslant T/M = \beta \geqslant \alpha + CT^{-1/3}$. Then $p_x(I \leqslant \alpha) < CT^{-1/6}$, C does not depend on x.*

Proof. From (5.1) and Lemma 2,

$$p_x(I \leqslant \alpha) \leqslant e^{-T} \cdot \sum \frac{\lambda^i}{i!} \cdot [T!\, T^{-T} e^T]$$
$$|i| = T, \qquad \sum_{k=0}^{M-1} i_k^2 \leqslant \alpha T + T. \quad (5.17)$$

Here $\lambda = (T/|X|)\, x = (\lambda_0, \ldots, \lambda_{M-1})$, $|\lambda| = T$. Consider a stochastic vector $\eta = (\eta_0, \ldots, \eta_{M-1})$ with the probability distribution

$$p(\eta_0 = i_0^2, \ldots, \eta_{M-1} = i_{M-1}^2)$$
$$= \frac{\lambda^i}{i!} e^{-T}[T!\, T^{-T} e^T] \qquad \text{if} \quad |i| = T,$$
$$= 0 \qquad \text{otherwise.} \quad (5.18)$$

Let

$$\varphi = \varphi(y_0, \ldots, y_{M-1}) = \sum_{|i|=T} \frac{\lambda_0^{i_0} \cdots \lambda_{M-1}^{i_{M-1}}\, y_0^{i_0} \cdots y_{M-1}^{i_{M-1}}}{i!} (T!\, T^{-T})$$
$$= e^{\lambda_0 y_0 + \cdots + \lambda_{M-1} y_{M-1}} \cdot T!\, T^{-T}.$$

To obtain the mathematical expectation $E\eta_K$, we take $y_K(\partial/\partial y_K)\, y_K(\partial/\partial y_K)\varphi$, then put $y_0 = \cdots = y_{M-1} = y$ and pick the coefficient at y^T out:

$$E\eta_K = T!\, T^{-T}\left(\lambda_K^2 \cdot \frac{T^{T-2}}{(T-2)!} + \frac{T^{T-1}}{(T-1)!}\lambda_K\right)$$
$$= \lambda_K^2 + \lambda_K - \frac{\lambda_K^2}{T}. \tag{5.19}$$

Likewise

$$E\eta_K^2 = T!\, T^{-T}\left(\lambda_K^4 \frac{T^{T-4}}{(T-4)!} + 6\lambda_K^3 \frac{T^{T-3}}{(T-3)!}\right.$$
$$\left. + 7\lambda_K^2 \frac{T^{T-2}}{(T-2)!} + \lambda_K \frac{T^{T-1}}{(T-1)!}\right)$$
$$\leqslant \lambda_K^4 + 6\lambda_K^3 + 7\lambda_K^2 + \lambda_K \tag{5.20}$$

and

$$E\eta_k\eta_l \leqslant (\lambda_K^2 + \lambda_K)(\lambda_l^2 + \lambda_l), \qquad k \neq l. \tag{5.21}$$

Next find the dispersion D of $\sum_{k=0}^{M-1} \eta_k$:

$$D = E\left(\sum_{k=0}^{M-1} (\eta_k - E\eta_k)\right)^2$$
$$= \sum_{k=0}^{M-1} (E\eta_k^2 - (E\eta_k)^2) + \sum_{k\neq l} (E\eta_k\eta_l - E\eta_k E\eta_l). \tag{5.22}$$

Using (5.19–(5.21), we transform (5.22) into the inequality

$$D \leqslant \sum_{k=0}^{M-1} (4\lambda_k^3 + 6\lambda_k^2 + \lambda_k) + \frac{2}{T}\sum_{k=0}^{M-1} (\lambda_k^4 + \lambda_k^3)$$
$$+ \frac{1}{T}\sum_{k\neq l} (\lambda_k^2\lambda_l^2 + \lambda_k^2\lambda_l)$$
$$\leqslant \sum_{k=0}^{M-1} (4\lambda_k^3 + 6\lambda_k^2 + \lambda_k) + \frac{2}{T}\left(\sum_{k=0}^{M-1} (\lambda_k^2 + \lambda_k)\right)^2. \tag{5.23}$$

Chebyshev inequality, (5.17), (5.18), and (5.19) yield

$$p_x(I \leqslant \alpha) \leqslant D \Big/ \left(\sum_{k=0}^{M-1} \lambda_k^2 - \alpha T - \frac{1}{T}\sum_{k=0}^{M-1} \lambda_k^2\right)^2. \tag{5.24}$$

Substitute the inequalities of the lemma and (5.23) into (5.24):

$$p_x(I \leqslant \alpha) \leqslant C \frac{(\sigma_2 - M\beta^2)^{3/2} + (\sigma_2 - M\beta^2)(\beta+1) + T\beta + T + T\beta^2 + (1/T)(\sigma_2 - M\beta^2)^2}{((\sigma_2 - M\beta^2) + T(\beta - \alpha) - (1/T)(\sigma_2 - M\beta^2) - \beta)^2} \tag{5.25}$$

where $\sigma_2 = \sum_{i=0}^{M-1} \lambda_i^2$. As it is known, $T\beta \leqslant \sigma_2 \leqslant T^2$ under the condition $\sum_{i=0}^{M-1} \lambda_i = T$. Hence,

$$\frac{1}{T}(\sigma_2 - M\beta^2)^2 \leqslant (\sigma_2 - M\beta^2)^{3/2}. \tag{5.26}$$

Rewrite the conditions of the lemma as

$$T(\beta - \alpha) \geqslant CT^{2/3}, \qquad \beta = O(T^{1/12}). \tag{5.27}$$

Two cases may present themselves, either $(\sigma_2 - M\beta^2)^{3/2} \leqslant T\beta^2$ or $(\sigma_2 - M\beta^2)^{3/2} > T\beta^2$. In the former (5.25)–(5.27) yield

$$p_x(I \leqslant \alpha) < C \frac{T\beta^2}{(T^{2/3} + o(T^{2/3}))^2} < CT^{-1/6}.$$

In the latter,

$$p_x(I \leqslant \alpha) < C \frac{(\sigma_2 - M\beta^2)^{3/2}}{(\sigma_2 - M\beta^2)^2(1 + o(1))} < \frac{1}{T^{1/3}\beta^{4/3}} < CT^{-1/6}. \quad \text{Q.E.D.}$$

6. Statement of the Theorem

Reminder: $\Delta(n, T)$ is the set of all dictionaries containing T n-length words; a-hash-set is a set of uniform maps from E^n to $[0, m-1]$ which for any $D \in \Delta(n, T)$ contains a map f such that $I(f, D) \leqslant a$; $N(n, T, \alpha, a)$ is the minimal cardinality of an a-hash-set given n, T, a, and $\alpha = T/m$; $L(n, T, \alpha, a)$ is the minimal program length of the worst map from an a-hash-set.

Theorem. *Let natural n, T, m tend to infinity in such a way that*

$$\frac{T^2}{2^n} \to 0, \qquad \lim \frac{\log T}{n} > 0, \qquad \frac{T}{m} = \alpha = \text{const.}$$

Then

(i) *As long as the index a is less than the loading factor α, the*

cardinality $N(n, T, \alpha, a)$ *and the complexity* $L(n, T, \alpha, a)$ *are very great. More precisely,*

$$C_2 T < \log N(n, T, \alpha, a) < C_1 T$$

$$C_4 T < L(n, T, \alpha, a) < C_3 T.$$

(ii) *No sooner the index a equals the loading factor, as both those numbers diminish essentially,*

$$C \log T < \log N(n, T, \alpha, \alpha) < \log T(1 + o(1))$$

$$C \log T < L(n, T, \alpha, \alpha) < n$$

(*the upper bound for* $L(n, T, \alpha, \alpha)$ *holds under the condition*: $\log m$ *is natural and is a divisor of* n).

(iii) *If* $a > \alpha$,

$$\log(n - 2 \log T)(1 + o(1)) < \log N(n, T, \alpha, a) < \tfrac{1}{2} \log T(1 + o(1))$$

$$\log(n - 2 \log T)(1 + o(1)) < L(n, T, \alpha, a) < \log m.$$

The upper bound for $L(n, T, \alpha, a)$ *holds under the conditions*: $\log m$ *is natural and is a divisor of* n, $\alpha = \alpha n / \log m$.

Note that the condition $\lim(\log T/n) > 0$ implies $\log T > Cn$, $n \to \infty$. Hence, the lower bounds (i) and (ii) equal the upper ones to withing constant factors and may be considered tight enough. It is not so with the bounds (iii).

7. Lower Bounds

(i) $a = \alpha - C$, $C > 0$. According to Lemma 3 with $\beta = \alpha$, $X = E^n$, $M = m$, we have $p_u(I \leqslant \beta - C) < e^{-C_1 T}$. It means that the index of any uniform map is less than $\alpha - c$ for $e^{-C_1 T}$th part of $\Delta(n, T)$ at the most. Therefore, the cardinality $N(n, T, \alpha, a)$ of an a-hash-set cannot be less than $(e^{-C_1 T})^{-1}$, and the lower bound (i) is proved.

(ii) We prove by reductio ad absurdum that $N(n, T, \alpha, \alpha) \geqslant m^{1/7}$, which is equivalent to the lower bound (ii). Let F be an α-hash-set for $\Delta(n, T)$ and suppose

$$|F| < m^{1/7}. \tag{7.1}$$

Number the members of F from 0 till $|F| - 1$, $F = \{f_0, \ldots, f_{|F|-1}\}$. We will develop inductively a sequence of sets $E^n = X^0 \supseteq X^1 \supseteq \cdots \supseteq X_{|F|}$. Suppose, the sets $X_0, \ldots, X_{i-1}$, $0 \leqslant i < |F|$ have already been determined. The set X_{i-1}

consists of, at most, m nonempty clusters with respect to the map f_i. The map f_i takes the words of a cluster to one and the same number. Take the largest cluster of X_{i-1}, then the largest one of the remainder, etc., $[m-m^{2/3}]$ clusters in all. Let X_i be the union of the clusters taken. If there are not so many clusters, let $X_i = X_{i-1}$. The number of those clusters is at least $(1/m)[m-m^{2/3}]$ part of all the clusters of X_{i-1}. Therefore, thanks to the way the clusters were chosen,

$$\frac{|X_i|}{|X_{i-1}|} \geqslant \frac{1}{m}(m-m^{2/3}) = 1-m^{-1/3}. \tag{7.2}$$

Use first (7.2) successively and then (7.1):

$$\frac{X_{|F|}}{|X_0|} \geqslant \frac{|X_1|}{|X_0|} \cdot \ldots \cdot \frac{X_{|F|}}{|X_{|F|-1}|} \geqslant (1-m^{-1/3})^{m^{1/7}} \geqslant \frac{1}{2}, \qquad m \to \infty. \tag{7.3}$$

Inequality (7.3) and the condition $\lim(T^2/2^n) = 0$ of the theorem imply that

$$\lim \frac{X_{|F|}}{T^2} = \infty. \tag{7.4}$$

Address Lemma 5 with $X_{|F|}$ playing the role of X. The range M of any $f_i \in F$ is $[m-m^{2/3}]$ at most. For the ratio β of T to M we have, on the one hand,

$$\beta = \frac{T}{M} \geqslant \frac{T}{m-m^{2/3}+1} \geqslant \alpha + Cm^{1/3}, \qquad \alpha = \frac{T}{m}. \tag{7.5}$$

On the other hand, all maps are uniform on E^n, and any cluster has contained $2^n/m$ words initially. Afterwards some words were deleted, hence for any f_i the set $X_{|F|}$ consists of no less than $|X_{|F|}| \cdot m/2^n$ clusters. It and (7.3) yield for β,

$$\beta = \frac{T}{M} \leqslant \frac{T \cdot 2^n}{|X_{|F|}| \cdot m} \leqslant 2\alpha. \tag{7.6}$$

Now we see from (7.4)–(7.6) that all conditions of Lemma 5 are met. Hence

$$p_x(I \leqslant \alpha) \leqslant CT^{-1/6}, \tag{7.7}$$

where $p_x(I \leqslant \alpha)$ is the number of dictionaries $D \subseteq X_{|F|}$, for which $I(f_i, D) < \alpha$, divided by $|X|$, x is the signature of f_i on X, $i = 0,\ldots,|F|-1$. The probability to meet a dictionary $D \subseteq X_{|F|}$ such that there is i_0, $0 \leqslant i_0 < |F|-1$, for which $I(f_{i_0}, D) < \alpha$ is no more than $|F| \cdot \max_x p_x(I < \alpha)$. As it follows from (7.1) and (7.7), this probability

tends to zero, $m \to \infty$. Therefore, there is a dictionary $D \subseteq X_{|F|}$ such that $I(D, f_i) > \alpha$ for all i, $0 \leqslant i \leqslant |F| - 1$, a contradiction, because F is an α-hash-set for $\Delta(n, T)$. Consequently, (7.1) is not true, and the lower bound (ii) is proved.

(iii) Given a and α choose a_1, $a_1 > a > \alpha$, and let

$$1 > \gamma = \frac{\alpha}{a_1}. \tag{7.8}$$

As it follows from the conditions $\lim_{n\to\infty}(\log T/n) > 0$ and $\lim(T^2/2^n) = 0$, there is a constant $C_1 > 2$ such that

$$T^{C_1} > 2^n. \tag{7.9}$$

Let F be an a-hash-set $F = \{f_0, \ldots, f_{|F|-1}\}$. We are going to prove that

$$|F| \geqslant \delta \frac{n - 2\log T}{\log(1/\gamma)}, \qquad \delta = \min\left(\frac{1}{2}, \frac{1}{12(C_1 - 2)}\right), \tag{7.10}$$

from which the claim (iii) follows. Suppose, on the contrary, that

$$|F| < \delta \frac{n - 2\log T}{\log(1/\gamma)}. \tag{7.11}$$

Develop a sequence $E^n = X_0 \supseteq X_1 \supseteq \cdots \supseteq X_{|F|}$ of sets, like in (ii). The set X_i consists either of $[\gamma m]$ largest clusters of X_{i-1}, if there are so many clusters in X_{i-1} with respect to f_i, or else $X_i = X_{i-1}$. Obviously,

$$X_{|F|} \geqslant 2^n \cdot \gamma^{|F|}. \tag{7.12}$$

The relations (7.11), (7.12), and $\lim(T^2/2^n) = 0$ yield

$$\lim \frac{|X_{|F|}|}{T^2} = \infty. \tag{7.13}$$

For the ratio β of T to the range of a map f_i, $0 \leqslant i \leqslant |F| - 1$, we obtain, on the one hand, from (7.8),

$$\beta \geqslant \frac{T}{\gamma m} = \frac{\alpha}{\gamma} = a_1 \geqslant \alpha + (a_1 - \alpha). \tag{7.14}$$

On the other hand, $X_{|F|}$ consists of no less than $X_{|F|} \cdot m/2^n$ clusters with respect to any $f_i \in F$. Hence, from (7.9), (7.11), and (7.12),

$$\beta \leqslant \frac{T \cdot 2^n}{m \cdot X_{|F|}} \leqslant \alpha T^{1/2}. \tag{7.15}$$

As it is seen from (7.13)–(7.15), all conditions of Lemma 5 are met. Therefore, $p_x(I \leqslant a) < CT^{-1/6}$. It yields, along with (7.11) and (7.9) that there is a dictionary $D \subseteq X$ on which all $f_i \in F$ have index more than α—a contradiction. Consequently, (7.11) does not hold, and the lower bound (iii) is proved.

The lower bounds for $L(n, T, \alpha, a)$ follow those for $N(n, T, \alpha, a)$ via (1.4).

8. Upper Bound for the Cardinalities of a-Hash-Sets

Upper bounds for $N(n, T, \alpha, a)$ will be proved by means of the following nice "covering lemma" from Nechiporuk (1965), see also Glagolev and Vasilev (1974). Let there be a set $\{K_1,..., K_l\}$, $l > 1$, of subsets of a set Δ such that any $x \in \Delta$ belongs to at least γl subsets. Then for any natural s there exist at most $s + (1-\gamma)^s |\Delta|$ subsets covering Δ, i.e., any x belongs to one of them.

Take the set $\Delta(n, T)$ of all dictionaries D, $|D| = T$, $D \subseteq E^n$. It is well known that if $T/2^n \to 0$, then

$$|\Delta(n, T)| = C_{2^n}^T = 2^{nH(2^n/T)(1+o(1))} = 2^{T(n-\log T)(1+o(1))}, \tag{8.1}$$

where $H(x) = -x \log x - (1-x)\log(1-x)$ is the entropy of x. Number from 1 till some l, $l > 1$, all uniform maps f which have

$$x_0 = \cdots = x_{m-1} = \frac{2^n}{m}, \qquad x_i = |f^{-1}(i)|, i = 0,..., m-1. \tag{8.2}$$

Fix a number a. Denote by K_i the set of all the dictionaries $D \subseteq \Delta(n, T)$ such that $I(f_i, D) \leqslant a$, $i = 1,..., l$. The probability to meet a function f such that $I(f, D) \leqslant a$ given a dictionary D equals the probability to meet a dictionary D, $I(f, D) \leqslant a$, given f, see Lemma 1. Hence, a dictionary D belongs to at least γl sets, where

$$\gamma = \min_{i=1,\ldots,l} p(I(f_i, D) \leqslant a). \tag{8.3}$$

Depending on whether $a < \alpha$, $a = \alpha$, or $a > \alpha$ we have different bounds for γ and, consequently, for $N(n, T, \alpha, a)$:

(i) $a < \alpha$. We restrict ourselves to a rather rough obvious bound

$$N(n, T, \alpha, a) \leqslant N(n, T, 1, 0), \qquad \alpha < 1. \tag{8.4}$$

The number $N(n, T, 1, 0)$ is the cardinality of a minimal set M such that for

any $D \in \Delta(n, T)$ there is an injection $f: D \to [0, T-1]$, $f \in M$. The upper bound

$$\log N(n, T, 1, 0) \leqslant T \log_2 e \tag{8.5}$$

is proved in Krichevsky (1978a) under the condition $\lim(\log T)/n) > 0$ which holds here. The upper bound (i) of the theorem follows (8.4) and (8.5).

(ii) $a = \alpha$. The second claim of Lemma 2 implies that for a uniform map $f: E^n \to [0, m-1]$, $(m/T)\sum_{k=0}^{m-1} i_k^2 = T + \chi^2$, where χ^2 is Pirson's χ^2-statistic. As it is shown in Medvedev (1970), χ^2-statistic under conditions of the theorem has asymtotically normal distribution. An immediate corollary of that is the inequality

$$\gamma > \tfrac{1}{2} - \varepsilon, \tag{8.6}$$

where ε is arbitrarily small, $n \to \infty$. The "covering lemma" with $s = \lceil (1/\gamma) \ln \gamma |\Delta| \rceil + 1$ yields, using (8.6) and (8.1),

$$N(n, T, \alpha, \alpha) \leqslant CT(n - \log T). \tag{8.7}$$

The upper bound (ii) for $N(n, T, \alpha, \alpha)$ follows (8.7) taking into account the condition $\log T/n > 0$, $n \to \infty$, of the theorem.

(iii) $a > \alpha$. Lemma 3(ii) provides a lower bound of γ:

$$\gamma \geqslant 1 - e^{-C\sqrt{T} \ln T}. \tag{8.8}$$

Next choose

$$s = C_1 \frac{\sqrt{T}}{\ln T}(n - \log T) \tag{8.9}$$

in the "covering lemma." We obtain, using (8.1) and (8.9), that the number of covering subsets is not greater than

$$\frac{C\sqrt{T}}{\ln T}(n - \log T). \tag{8.10}$$

The bound (9.10) takes the shape of inequality (iii) of the theorem, using the condition $\lim(\log T/n) > 0$ and its corollary $\lim(\log n/\log T) = 0$. Q.E.D.

9. Explicitly Constructed Hash-Sets; Hash-Functions Program Length Upperbounding

To be completed, the proof of the theorem needs upper bounds of $L(n, T, \alpha, a)$ only.

(i) $a < \alpha < 1$. An evident inequality

$$L(n, T, \alpha, a) \leqslant L(n, T, 1, 0)$$

which is similar to (8.4) is true. We use it with the inequality

$$L(n, T, 1, 0) \leqslant T \log e = T \cdot 1,4,...,$$

from (Krichevsky 1978b) and get just the upper bound (i) of the theorem for $L(n, T, \alpha, a)$.

(ii) K-linear hashing. Let $\log m = \mu$ be natural and a divisor of n, $g(x)$ be an irreducible polynomial of the degree μ with coefficients from $GF(2)$. Residues modulo $g(x)$ form the Galois field $GF(2^\mu)$ so that any μ-dimensional binary vector may be considered a μ-degree polynomial and, consequently, a member of $GF(2^\mu)$. For any n-dimensional vector y we denote by $y_1,..., y_{n/\mu}$ its successive μ-dimensional subvectors, so that the coordinates of y_1 are the first μ coordinates of y, etc. Any $b \in E^n$ determines a map $\varphi_b : E^n \to GF(2^\mu)$ through the formula

$$\varphi_b(x) = \sum_{i=1}^{n/\mu} b_i \cdot x_i. \tag{9.1}$$

Additions and multiplications are made in $GF(2^\mu)$. A member of $GF(2^\mu)$ may be considered a μ-dimensional vector and, hence, a natural number from $[0, m-1]$. Therefore, formula (9.1) provides a family Φ of maps $\varphi_b : E^n \to [0, m-1]$ of the cardinality

$$|\Phi| = 2^n. \tag{9.2}$$

Let for some different vectors $x' = (x'_1,..., x'_n)$, $x'' = (x''_1,..., x''_n)$ and a vector b

$$\varphi_b(x') = \varphi_b(x''). \tag{9.3}$$

Rewrite (9.3) using (9.1):

$$\sum_{i=1}^{n/\mu} b_i(x'_i + x''_i) = 0. \tag{9.4}$$

There exists i_0, $1 \leqslant i_0 \leqslant n/\mu$, such that $x'_{i_0} + x''_{i_0} \neq 0$ since x', x'' are different. Given any $(n/\mu) - 1$ vectors except the i_0th one, there is just one μ-

dimensional vector b_{i_0} satisfying (9.4). Therefore, there are $(2^\mu)^{(n/\mu)-1}$ different n-dimensional vectors $b = (b_1, ..., b_{n/\mu})$, satisfying (9.4). It means that for any $x, y \in E^n$, $\rho(\otimes \Phi x, \otimes \Phi y) \geqslant r$, where

$$r = 2^n - 2^{\mu(n/\mu-1)} = 2^n(1 - 2^{-\mu}) \tag{9.5}$$

and $\otimes \Phi$ is the tensor product of maps from the family Φ. Claim 1 provides any distionary D with a vector b such that the index of the map φ_b is not very great on D:

$$I(\varphi_b, D) \leqslant (|D| - 1)\left(1 - \frac{r}{|\Phi|}\right). \tag{9.6}$$

From (9.4), (9.5), and (9.6) we obtain

$$I(\varphi_b, D) \leqslant (|D| - 1)(1 - 1 + 2^{-\mu}) \leqslant \frac{T}{m} = \alpha.$$

Thus, α-hash-set with loading factor α is developed. Its cardinality is 2^n which is greater than the upper bound (ii) proved for $N(n, T, \alpha, \alpha)$ in Section 8. But, unlike the hash-set of Section 8, this hash-set is developed quite explicitly.

There are two parts in the program calculating a map φ_b. The first of them is just the vector b of n-bits length. The second, controlling part, consists of several computer instructions to find the hash-address of a word x according to formula (9.1). We are not going to write those instructions. But it is clear that they do not depend on the wordlength n. So, for the total length of the program $L(n, T, \alpha, \alpha)$ we obtain

$$L(n, T, \alpha, \alpha) \leqslant n + C = n(1 + o(1))$$

which is just the upper bound (ii).

(iii) $a > \alpha$. K-polynomial hashing. Let μ, $g(x)$, $x_1, ..., x_{n/\mu}$ be as in (ii). For a μ-dimensional vector b define the map $\varphi_b(x)$:

$$\varphi_b(x) = \sum_{i=1}^{n/\mu} x_i b^{i-1}, \tag{9.7}$$

additions and multiplications are made in $GF(2^\mu)$. Letting b be any vector of μ-dimension, we obtain by (9.7) a family Φ of maps from E^n to $[0, m-1]$ with the cardinality

$$|\Phi| = 2^\mu = m. \tag{9.8}$$

TABLE I

Complexity of Retrieval Algorithms

Retrieval algorithm	Program length in bits	Space used to compute in bits	Running time	Index I of hash-function. Average lists length equals $\frac{1}{2}I + 1$	Strong (s) or weak (w)
Search in an unordered table	Tn	Cn	CT	0	s
Search in a lexicographically ordered table	Tn	Cn	$C \log T$	0	s
Ordinary digital search	$CT \log T$	Cn	Cn	0	w
Enumerative algorithm	$T(n - \log T)$	$C \cdot 2^n$	$C \cdot 2^n$	0	s
Weak digital search (Krichevsky 1978b)	$CT \log n$	Cn	Cn	0	w
Strong digital search (Krichevsky 1978b)	$T(n - \log T)$	Cn	Cn	0	s
Weak search (Krichevsky 1978a)	$T \cdot \log_2 e \cdot \cdot (1 + \alpha^{-1}(1 - \alpha) \ln(1 - \alpha))$	Cn	Cn	0	w
K-linear hashing (Sect. 9ii)	n	Cn	Cn	α	w
K-polynomial hashing (Sect. 9iii)	$\log(T/\alpha)$	Cn	Cn	$\alpha n/\log m$	w

Notes. A dictionary D containing T n-length words is to be stored in an m-word table, $m = T/\alpha$. If the index equals zero, then each word of D occupies a separate position in the table. If the index is positive, then the words with the same address form a list. Lists are kept apart.

If $x' = (x'_1,..., x'_n) \neq x'' = (x''_1,..., x''_n)$ and for a vector b, $\varphi_b(x') = \varphi_b(x'')$, then

$$\sum_{i=1}^{n/\mu} (x'_i + x''_i)\, b^{i-1} = 0. \tag{9.9}$$

We get an algebraic equation for b of the degree $n/\mu - 1$. At least one its coefficient is not zero. Hence, there are at most $(n/\mu - 1)$ μ-dimensional vectors satisfying (9.9) and for any $x, y \in E^n$, $\rho(\otimes \Phi x, \otimes \Phi y) \geqslant r$, where

$$r = 2^{\mu} - \frac{n}{\mu} - 1, \tag{9.10}$$

$\otimes \Phi$ being the tensor product of maps from Φ. Claim 1 yields that for any $D \in \Delta(n, T)$ there is a vector $b = (b_1,..., b_\mu)$ such that $I(\varphi_b, D) \leqslant (|D| - 1)(1 - r/|\Phi|)$. From this, (9.8), and (9.10) we obtain

$$I(\varphi_b, D) \leqslant (T-1)\left(1 - 1 + \frac{n}{m \log m} + m^{-1}\right)$$

$$\leqslant \frac{Tn}{m \log m}(1 + o(1)) = \frac{\alpha n}{\log m}(1 + o(1)).$$

Thus, an α-hash-set of the cardinality m is developed, $a = (\alpha n/\log m)(1 + o(1))$ $(\log m = \log T + \log \alpha^{-1} < n$, since $\lim(T^2/2^n) = 0)$. The program for φ_b consists of the vector b and several computer instructions according to (9.7). Thus, $L(n, T, \alpha, (\alpha n/\log m)(1 + o(1)) < \log m$, which is the upper bound (iii). The proof of the theorem is completed.

ACKNOWLEDGMENTS

The author is very grateful to Mr. A. A. Borovkov, Mr. A. A. Mogulski, and Mr. S. V. Nagaev for their valuable help in proving Lemma 3; to the anonymous referee for his productive criticisms; and to Mrs. O. Korneeva for her aid in preparing the presentation.

REFERENCES

BERNSTEIN, S. N. (1946), "Probability Theory," Gostechizdat, Moscow. [Russian]

COVER, T. M. (1973), Enumerative source encoding *IEEE Trans. Inform. Theory* **IT-18**, No. 1, 73–77.

GLAGOLEV, V. V. AND VASILIEV, J. L. (1974), Algorithms to construct a minimal disjunctive normal form for a boolean function, *in* "Discteye Mathematics and Mathematical Problems of Cybernetics," pp. 67–98, Nauka, Moscow. [Russian]

JAESCHKE, S. (1981), Reciprocal hashing—a method for generating min. perfect hasing functions. *Comm. ACM*, **24**, No. 12, 829–833.

KNOTT, D. (1975), Hashing functions, *Comput. J.* **18**, No. 12, 265–278.

KNUTH, D. (1973), "The Art of Computer Programming," Vol. 3, Searching and sorting, Addison–Wesley, Reading, Mass.

KRICHEVSKY, R. (1978a), Optimum information search, Problems of Cybernetics No. 36, pp. 159–180, Nauka, Moscow.

KRICHEVSKY, R. (1978b), Digital enumeration of binary disctionaries, *Soviet Math. Dokl.* **19**, No. 2, 469–473.

LAVROV, S. S. AND GONCHAROVA, L. I. (1971), "Automatic Data Processing," Nauka, Moscow. [Russian]

MEDVEDEV, J. I. (1970), Some theorem on asymptotic distribution of χ^2-statistic, *Dokl. Acad. Sci. USSR*, **192**, No. 5, 987–990. [Russian]

NAGAEV, S. V. (1979), Large deviations of sums of independent random variables, *Ann. Probab.* **7**, No. 5, 745–789.

NECHIPORUK, E. I. (1965), On complexity of valve networks realizing partial boolean matrices, *Dokl. Akad. Sci. USSR*, **163**, No. 1, 40–43.

PETROV, V. V. (1972), Sums of independent random variables, Nauka, Moscow. [Russian]

Printed in Belgium